

133541US  
KYO-101

Title of the Invention

MULTI-FUNCTION PRINTER AND ITS CONTROL METHOD

Inventor

Makoto OYANAGI.

00973014-101101

## BACKGROUND OF THE INVENTION

### Field of the Invention

本発明は、マルチファンクションプリンタ、及び、その制御方法に関し、特に、コピー印刷における印刷時間の短縮化を図ったマルチファンクションプリンタ、及び、その制御方法に関する。

### Description of the Related Art

スキャナとプリンタとが一体化されて、1つの筐体に格納されたマルチファンクションプリンタが普及してきている。このようなマルチファンクションプリンタにおいては、1台で、スキャナとしての役割と、プリンタとしての役割と、コピー機としての役割とを、果たすことができる。この場合、プリンタ部分には、いわゆるページプリンタが用いられている。しかし、プリンタとして、いわゆるカラーインクジェットプリンタ等のシリアルプリンタを用いた方が、装置の小型化や低価格化を図ることができる。

ところで、コピー機としてこのマルチファンクションプリンタを使用する際には、スキャナでスキャンしたスキャンデータを、一時的に格納しておき、この格納したスキャンデータに基づいて、印刷イメージデータを生成している。そして、この印刷イメージデータをプリンタエンジンに転送し、印刷用紙に対する印刷を行っている。

しかし、プリンタとしてインクジェットプリンタ等のシリアルプリンタを用いる場合、印刷ヘッドを搭載したキャリッジの動作と紙送りの組み合わせにより、スキャナでスキャンしたスキャンデータと、印刷イメージデータとの間のデータ配列が異なる場合がある。その代表的な例がインターレース処理である。

図12は、このインターレース処理の概念を説明する図である。この図12の例では、このインクジェットプリンタの印刷ヘッドは、48個のインク吐出ノズルを備えており、したがって1回の印刷ヘッドの走査で48ライン分の印刷ができる。また、この例においては、2回の印刷ヘッドの走査で、1ラスタ分の印

刷ができる。すなわち、1回目の印刷ヘッドの走査で、#1、#3、…、#95からなる奇数ラインの印刷が行われ、2回目の印刷ヘッドの走査で、#2、#4、…、#96からなる偶数ラインの印刷が行われる。

なお、この図12の例では、1回目の印刷ヘッドの走査が終了した後に、1ライン分だけ紙送りをして、2回目の印刷ヘッドの走査を行う例を示したが、1回目の印刷ヘッドの走査が終了した後に、24ライン分（1/2ラスタ分）だけ紙送りして、2回目の印刷ヘッドの走査を行う場合もある。

さらに、スキャンデータと印刷イメージデータとで異なる配列になる代表的な例として、間引き印刷がある。この間引き印刷では、1ライン分のスキャンデータを、一定間隔に間引いて、高解像度印刷を行う。例えば、1ライン分のスキャンデータを2回の印刷ヘッドの移動で印刷を行う。

図13は、2回の印刷ヘッドの移動で1ライン分のスキャンデータを印刷する処理概念を説明する図である。この図13においては、#1のラインのみを示しているが、これ以外の#2～#96のラインについても同様である。

まず、1回目の印刷ヘッドの主走査で、スキャンデータの偶数ドットについて印刷を行う。続いて2回目の印刷ヘッドの主走査で、スキャンデータの奇数ドットについて印刷を行う。この2回目の印刷ヘッドの主走査による印刷では、1回目の印刷ヘッドの主走査で印刷した偶数ドットの間に、奇数ドットが位置するように、印刷を行う。

しかしながら、スキャンデータに基づいて印刷イメージデータを生成する処理には時間がかかり、印刷ヘッドを搭載したキャリッジ移動をする度にキャリッジが停止してしまうという問題があった。例えば、スキャンデータを偶数ドットと奇数ドットに振り分ける処理を、ソフトウェアを用いて行くと、その処理に相当の時間がかかってしまい、印刷ヘッドを搭載したキャリッジ移動をする度にキャリッジが停止してしまうという問題があった。つまり、1印刷パスごとにキャリッジ移動が停止してしまい、キャリッジを連続的に左右主走査方向に動作させることができないという問題があった。このため、スキャナで読み取ったスキャンデータを印刷しようとする際のパフォーマンスが低下してしまっていた。この問題は、CPUの処理能力が十分でない場合に、特に顕著な問題として発生してい

た。

## SUMMARY OF THE INVENTION

そこで本発明は、前記課題に鑑みてなされたものであり、スキャナで読み取ったスキャンデータを、異なるデータ配列の印刷イメージデータを用いて印刷を行うマルチファンクションプリンタにおいて、その印刷速度の高速化を図ることを1つの目的とする。

上記及び他の目的を達成するため、本発明における1つのアスペクトによれば、マルチファンクションプリンタは、スキャナとプリンタとが一体化されたマルチファンクションプリンタであって、前記スキャナで読み取ったスキャンデータを格納するための、第1データ格納部と、前記第1データ格納部に格納されているスキャンデータに基づいて、印刷処理に適したデータ形式である印刷イメージデータを生成して、この印刷イメージデータに基づいて、前記プリンタの印刷ヘッドを移動させながら、前記印刷ヘッドを駆動した印刷パスにより印刷を行う、印刷実行部と、前記スキャンデータを前記第1データ格納部に格納する際に、前記印刷イメージデータを生成するのに適した形式に前記スキャンデータを振り分けた上で、前記第1データ格納部に格納する、データ振り分け部と、を備える。

本発明における他のアスペクトによれば、マルチファンクションプリンタは、スキャナとプリンタとが一体化され、前記スキャナで読み取ったスキャンデータの同一ラインのデータを、X回の印刷ヘッドの主走査方向への駆動により前記プリンタで印刷することが可能なマルチファンクションプリンタであって、前記スキャンデータを、X回の印刷ヘッドの主走査方向の駆動に分けて印刷する際のそれぞれの回のデータ形式に対応するように振り分けた上で、第1データ格納部に格納する、振り分け格納部と、前記第1データ格納部から、振り分けられたスキャンデータを順番に読み出して、読み出す度にそのスキャンデータに基づいて印刷イメージデータを生成する、印刷イメージデータ生成部と、前記印刷イメージデータ生成部が生成した前記印刷イメージデータに基づいて、印刷ヘッドを主走査方向に駆動して印刷を行う、印刷実行部と、を備える。

本発明におけるさらなるアスペクトによれば、マルチファンクションプリンタ

の制御方法は、スキャナとプリンタとが一体化されたマルチファンクションプリンタの制御方法であって、前記スキャナで読み取ったスキャンデータを、実際に印刷をする際の印刷イメージデータを生成するのに適した形式に振り分ける工程と、振り分けた前記スキャンデータを、振り分けた状態で第1データ格納部に格納する工程と、前記第1データ格納部に格納されている前記スキャンデータに基づいて、印刷処理に適したデータ形式である印刷イメージデータを生成する工程と、前記印刷イメージデータに基づいて、前記プリンタの印刷ヘッドを駆動した印刷パスにより印刷を行う工程と、を備える。

本発明におけるそのさらなるアスペクトによれば、マルチファンクションプリンタの制御方法は、スキャナとプリンタとが一体化され、前記スキャナで読み取ったスキャンデータの同一ラインのデータを、X回の印刷ヘッドの主走査方向への駆動により前記プリンタで印刷することが可能なマルチファンクションプリンタの制御方法であって、前記スキャンデータを、X回の印刷ヘッドの主走査方向の駆動に分けて印刷する際のそれぞれの回のデータ形式に対応するように振り分ける工程と、振り分けた前記スキャンデータを、第1データ格納部に格納する工程と、前記第1データ格納部から、振り分けられたスキャンデータを順番に読み出して、読み出す度にそのスキャンデータに基づいて印刷イメージデータを生成する工程と、生成された前記印刷イメージデータに基づいて、印刷ヘッドを主走査方向に駆動して印刷を行う工程と、を備える。

#### BRIEF DESCRIPTION OF THE DRAWINGS

##### 【図1】

本発明の一実施形態に係るマルチファンクションプリンタの内部構成を示すブロック図である。

##### 【図2】

図1のマルチファンクションプリンタが備える各種のタスクを説明するブロック図である。

##### 【図3】

本実施形態に係るスキャナ処理（スキャナ処理タスク）の内容を説明するフロ

ーチャートである。

【図 4】

本実施形態に係るスキャナASICで実行される振り分け処理の内容を説明するフローチャートである。

【図 5】

図 4 に示す振り分け処理を実現するためのハードウェア構成の一例を示す図である。

【図 6】

本実施形態に係るインターレース展開処理（インターレース処理タスク）の内容を説明するフローチャートである。

【図 7】

偶数用インターレースメモリと奇数用インターレースメモリとから、インターレース処理された印刷イメージデータを生成する処理を説明する図である。

【図 8】

本実施形態に係る印刷実行処理（印刷実行処理タスク）の内容を説明するフローチャートである。

【図 9】

本発明の第 2 実施形態に係るスキャナASICで実行される振り分け処理の内容を説明するフローチャートである。

【図 10】

図 9 に示す振り分け処理で用いられる偶数用ルックアップテーブルの構成の一例を示す図である。

【図 11】

図 9 に示す振り分け処理で用いられる奇数用ルックアップテーブルの構成の一例を示す図である。

【図 12】

インターレース処理の処理概念を説明するための図である。

【図 13】

高解像度印刷で偶数ドットと奇数ドットを 2 回の印刷パスで印刷する場合を説

明する図である。

【図 1 4】

本発明の第 3 実施形態に係る高精細印刷の手法を説明する概念図である。

【図 1 5】

本発明の第 3 実施形態に係るインターレース展開処理（インターレース処理タスク）の内容を説明するフローチャートである。

【図 1 6】

第 3 実施形態に係る偶数用インターレースメモリに形成されるスキャンデータの構成を説明する図である。

【図 1 7】

第 3 実施形態に係る奇数用インターレースメモリに形成されるスキャンデータの構成を説明する図である。

【図 1 8】

本発明の第 3 実施形態における一変形例を説明する図である。

DETAILED DESCRIPTION OF THE INVENTION

〔第 1 実施形態〕

本発明の第 1 実施形態は、スキャンデータを A S I C を用いて偶数ビットと奇数ビットに振り分ける処理を行い、これら偶数ビットのスキャンデータと奇数ビットのスキャンデータとを、予めインターレースメモリ内に区分して格納しておくことにより、ソフトウェア処理として行われるインターレース処理の際に振り分け処理をしなくとも印刷イメージデータが生成できるようにしたものである。そして、これにより、可能な限り短い時間でスキャナで読み取ったスキャンデータを印刷しようとするものである。より詳しくを、以下に説明する。

まず、図 1 に基づいて、本実施形態に係るマルチファンクションプリンタ 5 の内部構成を説明する。この図 1 は、スキャナとプリンタとが一体化されたマルチファンクションプリンタ 5 の内部構成を示すブロック図である。

図 1 に示すように、マルチファンクションプリンタ 5 は、スキャナ機構部 1 0 と、スキャナ A S I C (Application Specific IC) 1 2 と、スキャナ用 R A M

(Random Access Memory) 14と、CPU (Central Processing Unit: 中央処理装置) 16と、プリンタ用RAM 18と、プリンタASIC 20と、プリンタエンジン22とを、備えている。

スキャナASIC 12と、スキャナ用RAM 14と、CPU 16と、プリンタ用RAM 18と、プリンタASIC 20とは、内部バスを介して相互に接続されている。スキャナ用RAM 14内には、スキャナ機構部10で読み取ったスキャンデータを一時的に格納するバッファ14aが生成されており、プリンタ用RAM 18内には、スキャンデータをインターレース処理するまで格納しておくインターレースメモリ18aが生成されている。本実施形態においては、スキャナ用RAM 14とプリンタ用RAM 18とを別個に設けているが、これらをまとめて1つのRAMとしてもよい。

また、本実施形態においては、インターレースメモリ18aは、偶数用インターレースメモリ24と、奇数用インターレースメモリ26とに区分されている。偶数用インターレースメモリ24には、スキャンデータのうち偶数ビットが格納され、奇数用インターレースメモリ26には、スキャンデータのうち奇数ビットが格納される。スキャンデータを偶数ビットと奇数ビットに振り分ける処理は、スキャナASIC 12で行われる。

スキャナ機構部10は、光学的に原稿を読み取るラインイメージセンサを有している。このラインイメージセンサは、キャリアッジに搭載されており、キャリアッジを原稿の一端側から他端側まで移動させることにより、原稿全体を読み取ることが可能である。この読み取り動作は、スキャナASIC 12が制御しており、読み取ったスキャンデータは、スキャナ用RAM 14内に生成されたバッファ14aに格納される。高解像度印刷の場合、所定量のスキャンデータがバッファ14aに蓄積された時点で、そのスキャンデータは、偶数ビットと奇数ビットに振り分けられた上で、それぞれ、プリンタ用RAM 18内に生成された偶数用インターレースメモリ24と奇数用インターレースメモリ26とに転送される。

偶数用インターレースメモリ24と奇数用インターレースメモリ26に格納されたスキャンデータは、CPU 16によりインターレース処理された上で、印刷イメージデータとして、プリンタASIC 20に送信される。本実施形態において



は、まず、偶数用インターレースメモリ24に格納されているスキャンデータに対してインターレース処理を施して印刷イメージデータを生成し、1印刷パス分の印刷イメージデータとして、プリンタASIC20に送信する。例えば、偶数用インターレースメモリ24の奇数ラインに格納されているスキャンデータに基づいて、1印刷パス分の印刷イメージを生成し、プリンタASIC20に送信する。プリンタASIC20は、この1印刷パス分の印刷イメージデータに基づいてプリンタエンジン22の制御を行い、印刷を行う。具体的には、印刷ヘッドを主走査方向に移動させながら、印刷ヘッドにある複数のインク吐出ノズルからインクを吐出して、印刷用紙に偶数ドットの印刷を行う。

続いて、CPU16は、奇数用インターレースメモリ26に格納されているスキャンデータに対してインターレース処理を施して印刷イメージデータを生成し、先ほど印刷した印刷パスと同じラインにある印刷パスの印刷イメージデータとして、プリンタASIC20に送信する。例えば、奇数用インターレースメモリ26の奇数ラインに格納されているスキャンデータに基づいて、1印刷パス分の印刷イメージを生成し、プリンタASIC20に送信する。プリンタASIC20は、印刷イメージデータに基づいてプリンタエンジン22の制御を行い、印刷を行う。この際、先ほど印刷用紙に印刷した偶数ドットの間に、今回印刷する奇数ドットが位置するように印刷をする。

次に、本実施形態に係るマルチファンクションプリンタ5は、1ライン分だけ、印刷用紙を副走査方向（主走査方向と交差する方向）に紙送りをする。そして、同様に、偶数用インターレースメモリ24の偶数ラインに格納されているスキャンデータを印刷し、奇数用インターレースメモリ26の偶数ラインに格納されているスキャンデータを印刷する。このように、印刷ヘッドを主走査方向に移動した印刷を4回行うことにより、1ラスタース分の印刷を行うことができる。

本実施形態では、このように1ライン上にあるスキャンデータを偶数ビットと奇数ビットに分けた上で、偶数ビットの印刷と奇数ビットの印刷を別個に行うとともに、奇数ラインと偶数ラインの印刷も個別に行うことにより、高解像度印刷を実現している。

以上のことからわかるように、本実施形態に係るマルチファンクションプリン

タ5は、ページ単位のメモリバッファを備えるのではなく、スキャンしたスキャンデータを、少なくとも1バンド単位（印刷ヘッドの高さ分）のプリンタ用RAM18に格納して、印刷を行う。

マルチファンクションプリンタ5の概略的处理内容は、以上の通りであるが、次に、マルチファンクションプリンタ5が備える各種のタスクについて説明する。

図2は、CPU16で処理される各種のタスクを示す図である。本実施形態においては、マルチファンクションプリンタ5に、中央処理装置としては1つのCPU16のみが設けられている。このため、このCPU16で、スキャナに関するタスクの処理とプリンタに関するタスクの処理との双方が行われることになる。また、本実施形態のマルチファンクションプリンタ5は、リアルタイムマルチタスクOS（オペレーティングシステム）30を採用している。このため、各種のタスクには、このリアルタイムマルチタスクOS30上で、CPU16が所定の優先順位で割り当てられることになる。

図2に示すように、本実施形態におけるマルチファンクションプリンタ5は、印刷実行処理タスク40と、スキャナ処理タスク41と、インターレース処理タスク42と、アイドルタスク43とを備えている。また、これ以外にも様々なタスクを、その他のタスク44として備えている。

各タスクの詳細な処理内容は後述するが、スキャナ処理タスク41は、上述したスキャン処理を行うためのタスクである。インターレース処理タスク42は、偶数用インターレースメモリ24及び奇数用インターレースメモリ26からスキャンデータを読み出して、インターレース処理を行うタスクである。印刷実行処理タスク40は、インターレース処理された印刷イメージデータに基づいて、印刷を行うタスクである。

図3は、本実施形態に係るスキャナ処理タスク41が行うスキャナ処理の内容を説明するフローチャートを示す図である。この図3に示すスキャナ処理は、インターレース処理タスク42から送信された転送要求により、起動される処理である。つまり、転送要求がトリガーとなって、スキャナ処理タスク41は起動する。この転送要求に際しては、スキャンデータを要求するライン数も指定されている。例えば、10ライン分のスキャンデータが必要である旨の指定がなされて

いる。

図3に示すように、スキャナ処理タスク41は、スキャナ機構部10のキャリアッジ移動用のモータを起動する（ステップS10）。そして、スキャナASIC12にスキャンの開始を指令する（ステップS11）。スキャン動作の具体的な制御はスキャナASIC12が行う。このため、スキャナ処理タスク41においては、スキャンの開始をスキャナASIC12に指示した後に、CPU16を解放する。

スキャナASIC12は、指定されたライン分のスキャン動作を行い、その読み取ったスキャンデータを、バッファ14aに格納する。例えば、10ライン分のスキャン指示を受信していた場合には、10ライン分のスキャンデータをバッファ14aに格納する。

続いて、スキャナASIC12は、バッファ14aに格納されたスキャンデータを、偶数ビットと奇数ビットに区分して、偶数ビットのスキャンデータを偶数用インターレースメモリ24に転送し、奇数ビットのスキャンデータを奇数用インターレースメモリ26に転送する。そして、バッファ14aに格納されているすべてのスキャンデータの転送を終えた時点で、スキャナASIC12は、スキャン終了の割り込みを発生する。

このスキャン終了の割り込みに基づいて、スキャナ処理タスク41は、再起動される。そして、図3に示すように、スキャナ処理タスク41は、転送が完了したことを示す転送完了通知を、インターレース処理タスク42に送信する（ステップS12）。これにより、スキャナ処理が終了する。

次に、図4及び図5に基づいて、スキャナASIC12で行われる上記スキャンデータの振り分け処理について詳しく説明する。この図4は、スキャナASIC12で行われる振り分け処理の内容を説明するフローチャートである。図5は、その振り分け処理を実現するために、スキャナASIC12内に設けられているハードウェア構成の一例を示す図である。

これら図4及び図5に示すように、バッファ14aに所定量のスキャンデータを格納した時点で、スキャナASIC12は、バッファ14aからスキャンデータをラッチし、ラッチバッファ50に格納する（ステップS20）。本実施形態

においては、16ビット単位（ワード単位）でこのラッチを行う。但し、ラッチするデータ長は、8ビット単位（バイト単位）や32ビット単位（ロングワード単位）等であってもよい。

次に、ビット $2n$ （ $n=0\sim7$ ）のスキャンデータを、偶数用インターレースメモリ24に転送し、ビット $2n+1$ （ $n=0\sim7$ ）のスキャンデータを、奇数用インターレースメモリ26に転送する（ステップS21）。これにより、ビット0、2、4、6、8、10、12、14のスキャンデータが、偶数用インターレースメモリ24に格納され、ビット1、3、5、7、9、11、13、15のスキャンデータが、奇数用インターレースメモリ26に格納される。

次に、偶数用インターレースメモリ24の格納先アドレスと、奇数用インターレースメモリ26の格納先アドレスとを、それぞれ、更新する（ステップS22）。格納先アドレスを更新することにより、次のスキャンデータを格納すべきアドレスが定まることになる。

次に、バッファ14aに格納されているすべてのスキャンデータを、偶数用インターレースメモリ24及び奇数用インターレースメモリ26に転送したかどうかを判断する（ステップS23）。すべてのスキャンデータを転送し終えた場合（ステップS23：Yes）には、この振り分け処理が終了する。上述したように、この際スキャナASIC12は、スキャン終了の割り込みを発生する。一方、すべてのスキャンデータを転送し終わっていない場合（ステップS23：No）には、上述したステップS20からの処理を繰り返す。

次に、図6及び図7に基づいて、インターレース処理タスク42の処理内容を説明する。この図6は、本実施形態に係るインターレース処理タスク42が行うインターレース展開処理の内容を説明するフローチャートを示す図である。この図6に示すインターレース展開処理は、スキャナ処理タスク41から送信された転送完了通知により、起動される処理である。つまり、転送完了通知がトリガーとなって、インターレース処理タスク42は起動する。図7は、偶数用インターレースメモリ24及び奇数用インターレースメモリ26に格納されているスキャンデータに基づいて、印刷イメージデータを生成する処理過程を説明する図である。

図6に示すように、本実施形態に係るインターレース処理タスク42は、まず、次の印刷パスで奇数ドットを印刷するのか、それとも偶数ドットを印刷するのかを、決定する（ステップS30）。続いて、インターレース処理タスク42は、次の印刷パスで奇数ドットを印刷すると決めたかどうかを判断する（ステップS31）。奇数ドットを印刷すると決めた場合（ステップS31：Yes）には、奇数用インターレースメモリ26に引き抜き用ポインタをセットする（ステップS32）。

例えば、図7において、奇数用インターレースメモリ26に格納されている奇数ライン#1、#3、…、#95の印刷をする場合には、引き抜き用ポインタを奇数用インターレースメモリ26の奇数ライン#1、#3、…、#95にセットする。つまり、本実施形態に係るマルチファンクションプリンタ5の印刷ヘッドは、48本のインク吐出ノズルを有している。

これに対して、図6に示すように、ステップS31で偶数ドットを印刷すると判断した場合（ステップS31：No）には、偶数用インターレースメモリ24に、引き抜き用ポインタをセットする（ステップS33）。

例えば、図7において、偶数用インターレースメモリ24に格納されている奇数ライン#1、#3、…、#95の印刷をする場合には、引き抜き用ポインタを偶数用インターレースメモリ24の奇数ライン#1、#3、…、#95にセットする。

次に、図6に示すように、インターレース処理タスク42は、引き抜き用ポインタからスキャンデータを取得して、印刷イメージデータを生成する（ステップS34）。本実施形態においては、図7に示すように、PD1～PD48からなる48ラインにより、1回の印刷パス分の印刷イメージデータを生成する。PD1～PD48は、それぞれ印刷ヘッドのインク吐出ノズル1～48に対応している。

次に、図6に示すように、インターレース処理タスク42は、生成した印刷イメージデータを印刷実行処理タスク40に送信する（ステップS35）。これにより、印刷ヘッドを主走査方向に1回移動した印刷が行われる。続いて、ページ管理カウンタを更新する（ステップS36）。このページ管理カウンタは、1ペ

ージ分の印刷イメージデータを生成したかどうかを判断するためのカウンタである。

続いて、インターレース処理タスク42は、このページ管理カウンタに基づいて、1ページ分のインターレース処理が終了したかどうかを判断する（ステップS37）。1ページ分のインターレース処理が終了した場合（ステップS37：Yes）には、このインターレース展開処理は終了する。

一方、1ページ分のインターレース処理が終了していない場合（ステップS37：No）には、インターレースメモリ18aに、次のインターレース処理を行うのに必要なスキャンデータが格納されているかどうかを判断する（ステップS38）。

次のインターレース処理を行うのに必要なスキャンデータが、インターレースメモリ18aに格納されていると判断した場合（ステップS38：Yes）には、上述したステップS30からの処理を繰り返す。

一方、次のインターレース処理を行うのに必要なスキャンデータが、インターレースメモリ18aに格納されていないと判断した場合（ステップS38：No）には、スキャナ処理タスク41へ、次の転送要求を送信する（ステップS39）。そして、このインターレース処理タスク42をひとまず終了する。この場合、上述したスキャナ処理タスク41から転送完了通知が発信されることにより、このインターレース処理タスク42は再起動される。

これら図6及び図7から分かるように、本実施形態においては、1ラスタース分のスキャンデータの印刷は、印刷ヘッドを4回主走査方向に移動することにより、行われる。例えば、1回目の印刷ヘッドの移動で、偶数用インターレースメモリ24の奇数ラインを印刷し、2回目の印刷ヘッドの移動で奇数用インターレースメモリ26の奇数ラインを印刷し、3回目の印刷ヘッドの移動で偶数用インターレースメモリ24の偶数ラインを印刷し、4回目の印刷ヘッドの移動で奇数用インターレースメモリの偶数ラインを印刷する。

次に、図8に基づいて、印刷実行処理タスク40の処理内容を説明する。この図8は、本実施形態に係る印刷実行処理タスク40が行う印刷実行処理の内容を説明するフローチャートを示す図である。この図8に示す印刷実行処理は、イン



る。図10及び図11は、図9の振り分け処理で用いられる偶数用ルックアップテーブルTB0と奇数用ルックアップテーブルTB1とをそれぞれ示す図である。

まず、図10及び図11に基づいて、偶数用ルックアップテーブルTB0と奇数用ルックアップテーブルTB1の構成について説明する。図10に示すように、偶数用ルックアップテーブルTB0には、8ビットのスキランデータすべてのパターンに対応する4ビットの偶数ビットデータが格納されている。つまり、8ビットのスキランデータにより発生し得る256パターンすべてについて、偶数ビットを抽出して得られる4ビットデータを、予め格納しておく。そして、8ビットのスキランデータを取得した際には、この偶数用ルックアップテーブルTB0を検索することにより、偶数ビットデータが得られるようにしておく。このことは、図11に示す奇数用ルックアップテーブルTB1についても同様である。本実施形態においては、これら偶数用ルックアップテーブルTB0と奇数用ルックアップテーブルTB1は、スキナASIC12内に形成されている。

次に図9に基づいて、スキナASIC12で行われる振り分け処理について説明する。この図9に示すように、バッファ14aに所定量のスキランデータを格納した時点で、スキナASIC12は、バッファ14aからスキランデータをラッチし、ラッチバッファ50に格納する（ステップS50）。本実施形態においては、8ビット単位（バイト単位）でこのラッチを行う。但し、ラッチするデータ長は、16ビット単位（ワード単位）や32ビット単位（ロングワード単位）等であってもよい。この場合、ラッチするデータ長に合わせて、偶数用ルックアップテーブルTB0と奇数用ルックアップテーブルTB1も、16ビットや32ビット用に形成しておく必要がある。

次に、偶数用ルックアップテーブルTB0を参照して、ステップS50でラッチしたスキランデータに対応する偶数ビットデータを取得し、この偶数ビットデータを偶数用インターレースメモリ24に転送する（ステップS51）。すなわち、図10に示す偶数用ルックアップテーブルTB0を参照し、8ビットのスキランデータから、偶数ビットを抽出した偶数ビットデータを取得する。例えば、ステップS50でスキランデータとして、「00110100」をラッチした場合、偶数用ルックアップテーブルTB0に基づいて、偶数ビットデータとして



「0110」を取得する。そして、この偶数ビットデータを偶数用インターレースメモリ24に転送する。

同様に、図9に示すように、次に奇数用ルックアップテーブルTB1を参照して、ステップS50でラッチしたスキャンデータに対応する奇数ビットデータを取得し、この奇数ビットデータを奇数用インターレースメモリ26に転送する（ステップS52）。すなわち、図11に示す奇数用ルックアップテーブルTB1を参照し、8ビットのスキャンデータから、奇数ビットを抽出した奇数ビットデータを取得する。例えば、上記と同様にステップS50でスキャンデータとして、「00110100」をラッチした場合、奇数用ルックアップテーブルTB1に基づいて、奇数ビットデータとして「0100」を取得する。そして、この奇数ビットデータを奇数用インターレースメモリ26に転送する。

次に、偶数用インターレースメモリ24の格納先アドレスと、奇数用インターレースメモリ26の格納先アドレスとを、それぞれ、更新する（ステップS53）。格納先アドレスを更新することにより、次のスキャンデータを格納すべきアドレスが定まることになる。

次に、バッファ14aに格納されているすべてのスキャンデータを、偶数用インターレースメモリ24及び奇数用インターレースメモリ26に転送したかどうかを判断する（ステップS54）。すべてのスキャンデータを転送し終えた場合（ステップS54：Yes）には、この振り分け処理が終了する。一方、すべてのスキャンデータを転送し終えていない場合（ステップS54：No）には、上述したステップS20からの処理を繰り返す。

以上のように、本実施形態に係るマルチファンクションプリンタ5によっても、上述した第1実施形態と同様に、スキャンデータの偶数ビットと奇数ビットとを、それぞれ、偶数用インターレースメモリ24と奇数用インターレースメモリ26とに振り分けて格納したので、インターレース処理タスク42ではスキャンデータを偶数ビットと奇数ビットに振り分ける必要がなくなる。このため、高解像度印刷であっても、プリンタエンジン22の最大スループットで印刷を行うことができる。プリンタのキャリッジ移動を停止させることなく、スキャナ機構部10で読み取ったスキャンデータを印刷することができるようになる。つまり、プリン

タの印刷ヘッドを搭載したキャリッジの主走査を停止させることなく、スキャナ機構部10で読み取ったスキャンデータを印刷することができるようになる。

また、本実施形態においても、スキャンデータを偶数ビットと奇数ビットとに振り分ける処理を、スキャナASIC12、つまりハードウェアで行うこととしたので、CPU16を1つしか備えていないマルチファンクションプリンタ5であっても、高速に処理することができる。特に、CPU16の処理速度が十分でない場合であっても、従来より短い印刷時間を実現することができる。

### 〔第3実施形態〕

本発明の第3実施形態は、上述した第1及び第2実施形態において、インターレース処理の仕方に變形を加えたものである。

図14は、本実施形態における高精細印刷の手法を説明する図である。この図14に示すように、本実施形態においては、スキャナ機構部10で読み取った1ライン分のスキャンデータを、偶数ドットと奇数ドットに分けて印刷するとともに、インターレース処理タスク42では3ラインのインターレース処理を行う。さらに、印刷された印刷用紙において矩形状に隣接する4つのドットが、異なるインク吐出ノズルで印刷されるように、主走査方向に印刷ヘッドを1回移動した後、副走査方向に印刷用紙を2ライン分だけ紙送りするようにしたものである。

すなわち、この図14の例では、印刷ヘッドは、0番から3番までの4つのインク吐出ノズルを有している。また、図14では、偶数ドットを丸印で表しており、奇数ドットを菱形印で表している。

1回目の印刷ヘッドの主走査方向への移動により、#2のラインの偶数ドットを3番ノズルで印刷し、2ライン分の紙送りをする。2回目の印刷ヘッドの主走査方向の移動により、#1と#4のラインの奇数ドットを2番ノズルと3番ノズルで印刷し、2ライン分の紙送りをする。3回目の印刷ヘッドの主走査方向の移動により、#3と#6のラインの偶数ドットを2番ノズルと3番ノズルで印刷し、2ライン分の紙送りをする。4回目の印刷ヘッドの主走査方向の移動により、#2と#5と#8のラインの奇数ドットを1番ノズルと2番ノズルと3番ノズルで印刷し、2ライン分の紙送りをする。5回目の印刷ヘッドの主走査方向の移動により、#1と#4と#7と#10のラインの偶数ドットを1番～4番のノズルで

印刷をする。

以下、このように、3ライン毎に、偶数ドットと奇数ドットとを交互に印刷する処理を繰り返して、スキャンデータの印刷を行っていく。但し、印刷用紙に正常に印刷できる有効印刷範囲は、図14に示すように、1回目の印刷ヘッドの移動における0番ノズルの位置から9番目のラインより、副走査方向下側になる。

また、印刷を行うために抜き出すスキャンデータのラインの間隔をK（図14の例では3）とし、紙送りのライン数をF（図14の例では2）とすると、KとFは互いに素の関係になっている。この関係を維持することにより、図14に点線で示すように、矩形状に隣接する4つのドットが互いに異なるインク吐出ノズルで印刷されるようになる。

本実施形態に係るマルチファンクションプリンタ5のハードウェア構成は、図1と同様である。また、第1実施形態又は第2実施形態で説明したように、スキヤナASIC12でスキャンデータが偶数ビットと奇数ビットに区分されて、それぞれ、偶数用インターレースメモリ24と奇数用インターレースメモリ26に格納されるのも同様である。しかし、CPU16が実行するインターレース処理タスク42の内容が異なっている。

図15は本実施形態に係るインターレース処理タスク42の処理内容を説明するフローチャートである。この図15に示すように、本実施形態においては、まず、偶数用インターレースメモリ24と奇数用インターレースメモリ26との先頭部分に、印刷開始用のダミーラインTDLを形成する（ステップS29）。図16は、偶数用インターレースメモリ24の構成を示す図であり、図17は、奇数用インターレースメモリ26の構成を示す図である。但し、これら偶数用インターレースメモリ24及び奇数用インターレースメモリ26は、1ページ分の容量をプリンタ用RAM18に一括で確保するのではなく、所定ライン分の容量をプリンタ用RAM18に確保した上で、印刷の終了したラインのスキャンデータは破棄することにより、順次使い回しがなされる。したがって、図16及び図17は、あくまでも理解を助けるための概念説明図である。

この図16に示すように、本実施形態に係るインターレース処理タスク42においては、スキャンデータの印刷を開始する前に、ライン#TDL1～ライン#

TDL 8からなるダミーラインTDLを偶数用インターレースメモリ 24の先頭に付加する。ダミーラインTDLを付加するのは、上述したように有効印刷範囲との関係で、印刷開始後の8ラインは、正常な印刷ができないためである。このため、インク吐出ノズルからインクの吐出をしないデータ、つまり、NULLデータを書き込んだダミーラインTDLを用意する。

同様に、図17に示すように、本実施形態に係るインターレース処理タスク42においては、スキャンデータの印刷を開始する前に、ライン#TDL 1～ライン#TDL 6からなるダミーラインTDLを奇数用インターレースメモリ 26の先頭に付加する。ダミーラインTDLを付加するのは、上述したように有効印刷範囲との関係で、印刷開始後の6ラインは、正常な印刷ができないためである。このため、インク吐出ノズルからインクの吐出をしないデータ、つまり、NULLデータを書き込んだダミーラインTDLを用意する。但し、これらインターレースメモリ 24、26にダミーラインTDLをセットするのは、そのコピー印刷を開始する際の1回だけである。

次に図15に示すように、インターレース処理タスク42は、次の印刷パスで奇数ドットを印刷するのか、それとも偶数ドットを印刷するのかを、決定する

(ステップS30)。続いて、インターレース処理タスク42は、次の印刷パスで奇数ドットを印刷すると決めたかどうかを判断する(ステップS31)。偶数ドットを印刷すると決めた場合(ステップS31: No)には、偶数用インターレースメモリ 24に引き抜き用ポインタをセットする(ステップS33A)。

この引き抜きポインタのセットの仕方は、上述した第1及び第2実施形態と異なる。すなわち、図16に示すように、偶数用インターレースメモリ 24の先頭から、3行おきに引き抜きポインタをセットする。すなわち、ダミーライン#TDL 1から3行おきに引き抜きポインタをセットする。但し、次の引き抜きポインタをセットする際には、前回にセットした引き抜きポインタから4ライン分、副走査方向にシフトすることとなる。

一方、奇数ドットを印刷すると決めた場合(ステップS31: Yes)には、奇数用インターレースメモリ 26に引き抜き用ポインタをセットする(ステップS32A)。

この引き抜きポイントのセットの仕方は、上述した偶数用インターレースメモリ24の場合と同様である。すなわち、図17に示すように、奇数用インターレースメモリ26の先頭から、3行おきに引き抜きポイントをセットする。すなわち、ダミーライン#TDL1から3行おきに引き抜きポイントをセットする。但し、次の引き抜きポイントをセットする際には、前回にセットした引き抜きポイントから4ライン分、副走査方向にシフトすることとなる。

次に、図15に示すように、インターレース処理タスク42は、引き抜き用ポイントからスキャンデータを取得して、印刷イメージデータを生成する（ステップS34）。そして、インターレース処理タスク42は、生成した印刷イメージデータを印刷実行処理タスク40に送信する（ステップS35）。これにより、印刷ヘッドを主走査方向に1回移動した印刷が行われる。

次に、ページ管理カウンタを更新する（ステップS36）。このページ管理カウンタは、1ページ分の印刷イメージデータを生成したかどうかを判断するためのカウンタである。

続いて、インターレース処理タスク42は、このページ管理カウンタに基づいて、1ページ分のスキャンデータの受信が終了したかどうかを判断する（ステップS37A）。1ページ分のスキャンデータの受信が終了した場合（ステップS37A：Yes）には、偶数用インターレースメモリ24と奇数用インターレースメモリ26に、印刷終了用のダミーラインBDLをセットする（ステップS37B）。

図16の例では、スキャンデータの印刷を終了する前に、ライン#BDL1～ライン#BDL8からなるダミーラインBDLを偶数用インターレースメモリ24の末尾に付加する。ダミーラインBDLを付加するのは、上述したように有効印刷範囲との関係で、印刷終了直前の8ラインは、正常な印刷ができないためである。このため、インク吐出ノズルからインクの吐出をしないデータ、つまり、NULLデータを書き込んだダミーラインTDLを用意する。

同様に、図17に示すように、スキャンデータの印刷を終了する前に、ライン#BDL1～ライン#BDL6からなるダミーラインBDLを奇数用インターレースメモリ26の末尾に付加する。ダミーラインBDLを付加するのは、上述し

たように有効印刷範囲との関係で、印刷終了直前の6ラインは、正常な印刷ができないためである。このため、インク吐出ノズルからインクの吐出をしないデータ、つまり、NULLデータを書き込んだダミーラインTDLを用意する。

次に、図15に示すように、インターレース処理タスク42は、ダミーラインBDLの最後まで印刷を行う（ステップS37C）。すなわち、図16に示す偶数用インターレースメモリ24においては、3ラインおきに引き抜きポイントをセットしながら、1回の印刷パスが終わる毎に4ライン分シフトして、ダミーライン#BDL8まで印刷を行う。これにより、スキャンデータの最後のライン#nまで正常な印刷結果を得ることがができる。

また、図17に示す奇数用インターレースメモリ26においては、3ラインおきに引き抜きポイントをセットしながら、1回の印刷パスが終わる毎に4ライン分シフトして、ダミーライン#BDL8まで印刷を行う。これにより、スキャンデータの最後のライン#nまで正常な印刷結果を得ることがができる。これにより、インターレース処理タスク42の処理が終了する。

一方、図15に示すように、1ページ分のスキャンデータが終了していない場合（ステップS37A：No）には、インターレースメモリ18aに、次のインターレース処理を行うのに必要なスキャンデータが格納されているかどうかを判断する（ステップS38）。

次のインターレース処理を行うのに必要なスキャンデータが、インターレースメモリ18aに格納されていると判断した場合（ステップS38：Yes）には、上述したステップS30からの処理を繰り返す。

一方、次のインターレース処理を行うのに必要なスキャンデータが、インターレースメモリ18aに格納されていないと判断した場合（ステップS38：No）には、スキャナ処理タスク41へ、次の転送要求を送信する（ステップS39）。そして、このインターレース処理タスク42をひとまず終了する。この場合、上述したスキャナ処理タスク41から転送完了通知が発信されることにより、このインターレース処理タスク42はステップS30から再起動される。

以上のように、本実施形態に係るマルチファンクションプリンタ5によっても、上述した第1及び第2実施形態と同様に、スキャンデータの偶数ビットと奇数ビ

ットとを、それぞれ、偶数用インターレースメモリ24と奇数用インターレースメモリ26とに振り分けて格納したので、インターレース処理タスク42ではスキヤンデータを偶数ビットと奇数ビットに振り分ける必要がなくなる。このため、高解像度印刷であっても、プリンタエンジン22の最大スループットで印刷を行うことができる。プリンタのキャリッジ移動を停止させることなく、スキヤン機構部10で読み取ったスキヤンデータを印刷することができるようになる。つまり、プリンタの印刷ヘッドを搭載したキャリッジの主走査を停止させることなく、スキヤン機構部10で読み取ったスキヤンデータを印刷することができるようになる。

また、本実施形態においても、スキヤンデータを偶数ビットと奇数ビットとに振り分ける処理を、スキヤナASIC12、つまりハードウェアで行うこととしたので、CPU16を1つしか備えていないマルチファンクションプリンタ5であっても、高速に処理することができる。特に、CPU16の処理速度が十分でない場合であっても、従来より短い印刷時間を実現することができる。

なお、本発明は上記実施形態に限定されず種々に変形可能である。例えば、上述した各実施形態においては、1ドットの印刷イメージデータを、1ビットのデータから生成することとしたが、これに限るものではない。すなわち、1ドットの印刷イメージデータを、2ビット等（例えば、00、01、10、11）の多値データに基づいて生成するようにしてもよい。この場合、1つのドットについて、大ドット、中ドット、小ドット、ドット無しの4つのパターンが存在することになる。

また、上述した各実施形態においては、スキヤンデータの1つのラインを2回の印刷パスで印刷する場合を説明したが、スキヤンデータの1つのラインを3回、4回…の印刷パスで印刷する場合でも、本発明を適用することができる。この場合、これに合わせて、インターレースメモリ18aを、3個、4個…に区分し、これに合わせてスキヤナASIC12がスキヤンデータを振り分けるようにすればよい。

さらに、上述した各実施形態においては、マルチファンクションプリンタ5は印刷用紙に印刷されたラインとラインとの間を、別のラインで再び印刷するイン

ターレース処理を行っているが、このインターレース処理は本発明において必ずしも必要なものではない。つまり、インターレース処理をしないマルチファンクションプリンタについても、本発明を適用することができる。

さらに、上述した実施形態においては、スキャンデータの振り分け処理を、スキャナASIC12で構成したが、これを他のハードウェア機構又はソフトウェア機構で構成してもよい。

さらに、高解像度印刷をする仕組みは、上述した実施形態に限るものではない。印刷用紙上の解像度よりも印刷ヘッドのインク吐出ノズルの解像度の方が粗く、複数回の印刷パスで印刷用紙上の印刷が完成する仕組みのプリンタであれば、本発明を適用することができる。

また、上述した実施形態においては、スキャンデータに基づいて、プリンタエンジン22で印刷可能な印刷データを生成するための処理が、高解像度印刷のためのビット振り分け処理である場合を説明したが、スキャンデータに対して他の処理を施すことによりプリンタエンジン22で印刷可能な印刷データを生成する場合でも、本発明を適用することができる。例えば、第3実施形態において、図18に示すように、3ライン毎にスキャンデータを取り出すインターレース処理をした上で、スキャナASIC12でスキャンデータを偶数ビットと奇数ビットとに分けるとともに、プリンタ用RAM18に格納するようにしてもよい。このようにすれば、プリンタ用RAM18には、印刷パスの順にスキャンデータが格納されることとなる。このため、印刷の際にはCPU16では、インターレース処理も不要になり、プリンタ用RAM18の上から4ラインずつ、スキャンデータを取り出して、印刷を行えば足りるようになる。

また、別な例としては、上述した第1乃至第3実施形態において、偶数ビットと奇数ビットとを抜き出す処理を行わず、インターレース処理だけを行うマルチファンクションプリンタに対しても、本発明を適用することができる。この場合、例えば、第3実施形態においては、3ライン毎にスキャンデータを取り出すインターレース処理をした上で、プリンタ用RAM18に格納するようにしてもよい。このようにすれば、マルチファンクションプリンタは、印刷の際にプリンタ用RAM18の上から順に4ラインずつ、スキャンデータを取り出して印刷を行えば



足りる。なお、この場合、3ライン毎にスキャンデータを取り出すインターレース処理のための振り分けは、スキャナASIC12で行ってもよいし、CPU16で行ってもよい。

さらに、上述した実施形態においては、同一ラインの中で偶数ビットを奇数ビットよりも先に印刷するようにしたが、これを逆にして、奇数ビットを偶数ビットよりも先に印刷するようにしてもよい。

また、上述した実施形態においては、印刷媒体が印刷用紙である場合を例に説明したが、OHPシート等の他の印刷媒体であってもよい。

また、上述の実施形態で説明した印刷実行処理タスク40、スキャナ処理タスク41、及び、インタレース処理タスク42等の各タスク処理については、これら各処理を実行するためのプログラムをフロッピーディスク、CD-ROM (Compact Disc-Read Only Memory)、ROM、メモ리카ード等の記録媒体に記録して、記録媒体の形で頒布することが可能である。この場合、このプログラムが記録された記録媒体をマルチファンクションプリンタ5に読み込ませ、実行させることにより、上述した実施形態を実現することができる。

また、マルチファンクションプリンタ5は、オペレーティングシステムや別のアプリケーションプログラム等の他のプログラムを備える場合がある。この場合、マルチファンクションプリンタ5の備える他のプログラムを活用し、記録媒体にはそのマルチファンクションプリンタ5が備えるプログラムの中から、上述した実施形態と同等の処理を実現するプログラムを呼び出すような命令を記録するようにしてもよい。

さらに、このようなプログラムは、記録媒体の形ではなく、ネットワークを通じて搬送波として頒布することも可能である。ネットワーク上を搬送波の形で伝送されたプログラムは、マルチファンクションプリンタ5に取り込まれて、このプログラムを実行することにより上述した実施形態を実現することができる。

また、記録媒体にプログラムを記録する際や、ネットワーク上を搬送波として伝送される際に、プログラムの暗号化や圧縮化がなされている場合がある。この場合には、これら記録媒体や搬送波からプログラムを読み込んだマルチファンクションプリンタ5は、そのプログラムの復号化や伸張化を行った上で、実行する

必要がある。

09973914-101101